Remarks:

Reconsideration of the above referenced application in view of the enclosed amendment and remarks is requested. Claims 22 and 25 have been amended. Existing Claims 1 to 28 remain in the application.

ARGUMENT

Claims 1, 6 and 7 are rejected under 35 U.S.C. § 102(e) as being unpatentable over U.S. Patent No. 6,718,286 to Rivin et al. (hereinafter "Rivin et al."). This rejection is respectfully traversed and Claims 1, 6 and 7 are believed allowable based on the following discussion.

Rivin et al. teach a system for application code profiling. The profiling comprises retrieving the program counter, or equivalent, at random intervals of time in order to determine how much of the execution time is spent in each module or subroutine. Rivin et al. is able to perform the profiling "non-intrusively" because the program counter is retrieved using a JTAG register. The retrieved instruction counter is retrieved by a sampler (Fig. 1, 18), from the digital signal processor (DSP) (Fig. 1, 10), via the JTAG register ((Fig. 1, 16). At no time do Rivin et al. teach that any other data can be sampled from the DSP.

In contrast, Applicants' claimed invention comprises *gathering a first data sample during execution of a program; executing the program during a random inter-sample period; and gathering a second data sample following the inter-sample period*. The data sample as recited in Applicants' claims in not merely a program counter or instruction address, as taught by Rivin et al. Applicants define what is meant by sampling data and instrumentation in the Background section as originally filed:

> "Sampling" refers to obtaining data samples from an instrument that is performing a task, for example, a computer processor executing an application. "Instrumentation" refers to specific pieces of hardware that are used to record events, e.g., counters, registers and memory devices that store values reflecting the occurrence of events caused by the execution of the application.
>
> Generally, to obtain a sample of data, a sampling program interrupts the application being executed by the computer processor and then executes the sampling program to obtain a data sample. The sampling program is executed several times to

obtain a set of data samples. The set of data samples is used to determine the performance of the instrument while performing the application, for example, the level of memory utilization of a computer processor while executing a simulation application." (Emphasis added) (Background, page 1).

It will be apparent to one of ordinary skill in the art that sampling data to determine, for instance, memory utilization of the processor cannot be performed by the system as taught by Rivin et al. Rivin et al. do not teach collection of data samples, but only collection of random instruction counter (program counter (PC)) information.

Further, Rivin et al. teach retrieving the PC at random intervals. The PC information is to typically be analyzed off line, perhaps on an external host computer (Col. 2, lines 50, et seq.). Rivin et al. collects the PC via the JTAG register. Applicants' claimed invention requires that the data samples be gathered in-between *random inter-sample periods*. It is not the sampling that is random, but the periods between when sampling occurs that is random.

Applicants' invention uses the inter-sample period to perform any overhead involved in analyzing the data, or controlling the sample gathering periods. During the periods between the inter-sample periods, data samples are gathered. Thus, the overhead is performed at a time when samples are not being gathered. Moreover, Applicants' claimed invention recites that the program executes throughout the sample and non-sample (inter-sample) periods, but that data samples are gathered only at intervals between, or before and after the inter-sample periods.

With regard to Claim 6, the Examiner asserts that the SAMPLE and DISABLE signals reset the hardware and that sampling of the signal is stopped through removal of the signal. The DISABLE signal is not used to stop the hardware. The DISABLE signal is used to ensure that the contents of the JTAG register do not change if the PC changes while the JTAG register is being read. Neither of these signals reset any hardware. The SAMPLE signal merely allows the contents of the register (JTAG) to be retrieved by the sampler. The signals are sent at random intervals to collect, not to reset the hardware. The hardware continues to operate at all times. It is the collection of individual register contents (PC) that is enabled by the SAMPLE signal. This is not equivalent to a hardware reset.

Similarly, for Claim 7, Rivin et al. do not teach *starting* data gathering hardware and *stopping* the data gathering hardware. At most, Rivin et al. start and stop the actual collection of

the PC register. However, at no time do Rivin et al. teach the start (or reset) or stop of the hardware. Moreover, Rivin et al. do not teach *data gathering* as defined by Applicants. Rivin et al. only teach collection of the PC information.

Claims 20, 25 and 26 are rejected under 35 U.S.C. § 102(a) as being anticipated by Applicants' Background description. This rejection is respectfully traversed and Claims 20, 25 and 26 are believed allowable based on the foregoing and following discussion.

The Examiner misunderstands the advantages of the inter-sample period, as recited in Applicants' claims. The Examiner asserts that the discussion of interrupting the application being executed to execute the sampling program is the same as gathering data before, after and in-between inter-sampling periods. Applicants' claimed invention requires that a first data sample and a second data sample are separated by an inter-sample period. Applicants are not reducing sampling overhead by reducing the sample period; this method tends to result in calculating only average values. Instead, Applicants move the overhead to a period where no samples are collected at all, i.e., the inter-sample period. Thus, the first and second samples gathered more accurately reflect the actual performance if the machine. Nothing in the discussion of the Background implies that the method of interrupting the application multiple times to take data samples will result in more accurate samples. Applicants' claimed invention requires a specific period when no samples are collected (i.e., inter-sample period), not just that samples are collected at various intervals.

Regarding Claims 25-26, the Examiner likens the "duration of execution of the sampling program" to an explicit start/stop/reset of data gathering hardware. However, nothing in the Background states or implies that data gathering hardware is started or stopped. Applicants' require that the data gathering hardware is stopped to *commence the inter-sample period.* Further, if the data gathering hardware is started after an inter-sample period, it ends the inter-sample period. Nothing in the Background states or implies that an inter-sample period is used to separate sample data gathering periods, or the advantages of doing so.

Claims 2-5 and 8 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Rivin et al. and further in view of U.S. Patent No. 5,768,500 to Agrawal et al. (hereinafter "Agrawal et al.") This rejection is respectfully traversed and Claims 2-5 and 8 are believed allowable based on the foregoing and following discussion.

The Examiner likens the cache miss counter to the inter-sample counter as recited by Applicants. Applying the cache miss counter as taught by Agrawal et al. to the teachings of Rivin et al. will, at most, result in collecting program counter information when a predetermined number of cache misses have occurred. If there are no cache misses, then no information will be collected. In contrast, Applicants recite *generating an inter-sample count* which is then decremented using a clock pulse. It will be apparent to one of ordinary skill in the art that a predetermined count is not equivalent to *generating an inter-sample count* as recited in Claims 2-5 and 8. Further, the inter-sample count is decremented with the aid of a clock pulse, thereby being related to a temporal effect. Agrawal et al. do not teach or suggest a temporal delay, but only teach that after a certain number of cache misses occur that an interrupt is initiated to collect data.

Regarding Claim 3, the Examiner asserts that Rivin et al. teach performing overhead operations during the inter-sample period. Rivin et al. teach that the host may call for varying sampling intervals separated by non-sampling intervals (Col. 6, lines 50-53). However, Col. 6, lines 54-60), as cited by the Examiner, merely describes that the sampled data may be exported to a host via an in-circuit emulator and be manipulated by common statistical analysis software. At no time do Rivin et al. teach that analysis, or other, operations are performed during the inter-sample period. In fact, the host taught by Rivin et al. can easily perform analysis during a sampling period, which is contrary to Applicants' claimed invention.

With regard to Claims 4-5, the Examiner misunderstands the overall combination of elements of the claimed invention. First, the claimed data sample is not limited to a program counter and may comprise more than the reading of one register. The Examiner asserts that the sequential storing as taught by Rivin et al. means that storing before sampling again is inherent in these teachings. However, Applicants' gathered data samples may require more than just one read or store cycle. Rivin et al. does not teach an explicit inter-sample period which leaves time to perform overhead operations by being *longer* than the timer required to perform the data operations. In fact, there is no requirement that there is any lag time between the sequential retrieval of processor instruction as taught by Rivin et al. Thus, there is no guarantee that the lag time is longer than necessary. It could be just enough time.

With regard to Claim 8, combination of the cache miss counter of Agrawal et al. with the PC retrieval as taught by Rivin et al. will not result in Applicants' claimed invention, as described above.

Claims 9 and 10 are rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Rivin et al. and Agrawal et al. and further in view of U.S. Patent No. 3,700,869 to Low et al. (hereinafter "Low et al.") This rejection is respectfully traversed and Claims 9-10 are believed allowable based on the foregoing and following discussion.

Low et al. teach a feedback shift register. Rivin et al. and Agrawal et al. (alone or in combination) do not teach the other elements of Applicants' claimed invention. For instance, at least, generating an inter-sample count is not taught by the cited references. Thus, combining the feedback shift register as taught by Low et al. with the other cited references will not result in Applicants' claimed invention.

Claims 11 and 16 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Rivin et al. in view of U.S. Patent No. 6,070,009 to Dean et al. (hereinafter "Dean et al.") This rejection is respectfully traversed and Claims 11 and 16 are believed allowable based on the foregoing and following discussion.

Dean et al. teach storing various instruction sequences for estimating execution rates of program execution paths onto a machine-readable medium. However, combining the teachings of Dean et al. that instruction sequences may be stored on a machine-readable medium with the teachings of Rivin et al. will not result in Applicants' invention, as discussed above. Rivin et al. fails to teach the other recited elements of Applicants' claimed invention.

Claims 12-15 and 17 are rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Rivin et al. and Dean et al. and further in view of Agrawal et al. This rejection is respectfully traversed and Claims 12-15 and 17 are believed allowable based on the foregoing discussion, especially with respect to Claims 2-5, 8, 11 and 16.

Claims 18-19 are rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Rivin et al., Dean et al. and Agrawal et al. and further in view of Low et al. This rejection is respectfully traversed and Claims 18-19 are believed allowable based on the foregoing discussion, especially with respect to Claims 9-11 and 16.

Claim 21 is rejected under 35 U.S.C. § 103(a) as being unpatentable over the Background section and further in view of Agrawal et al. This rejection is respectfully traversed and Claim 21 is believed allowable based on the foregoing discussion, especially with respect to Claims 11, 16, 20, 25 and 26.

Claims 22-24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of the Background section and Agrawal et al. and further in view of "interrupts" by Daqarta (hereinafter "Daqarta"). This rejection is respectfully traversed and Claims 22-24 are believed allowable based on the foregoing and following discussion.

Daqarta does not teach or suggest an inter-sample period as described and claimed by Applicants. Daqarta teaches running an interrupt handler to transfer sample data to memory when the sample data is ready. The Examiner asserts that the inter-sample period is inherent in the teachings of Daqarta. However, Daqarta teaches, and the Examiner admits, that execution of the application stops during interrupt processing. Applicants' require that the application is running during the inter-sample period. Further, Daqarta does not teach an inter-sample period as described and claimed by Applicants.

With regard to Claim 23, the Examiner contradicts the argument regarding Claim 22, as incorporated, by asserting that inter-sample time longer than overhead execution is inherent. For Claim 22, the Examiner argued that an inter-sample period is inherent in Daqarta because the system has been interrupted while the overhead takes place. Now, the Examiner asserts that the application is also executing during the inter-sample period. It cannot be argued both ways. Further, as the inter-sample period is described as randomly generated in some embodiments, it may be as long a time as the overhead processing requires at one period and a longer time than the overhead processing during another period. Generating an inter-sample count is not taught or suggested by any of the references either alone or in combination. Thus, Claim 23 is believed allowable. Similarly, Claim 24 is believed allowable for the reasons discussed above.

Claims 27-28 are rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of the Background section and Agrawal et al. and further in view of Low et al. This rejection is respectfully traversed and Claims 27-28 are believed allowable based on the foregoing and following discussion.

Similar to the discussion above for Claims 9 and 10, Low et al. teach a feedback shift register. The Background section and Agrawal et al. (alone or in combination) do not teach the other elements of Applicants' claimed invention. For instance, at least, generating an inter-sample count is not taught by the cited references or by the Background. Thus, combining the feedback shift register as taught by Low et al. with the other cited references will not result in Applicants' claimed invention. All claims remaining in the application are now allowable.

CONCLUSION

In view of the foregoing, Claims 1 to 28 are all in condition for allowance. If the Examiner has any questions, the Examiner is invited to contact the undersigned at (703) 633-6845. Early issuance of Notice of Allowance is respectfully requested. Please charge any shortage of fees in connection with the filing of this paper, including extension of time fees, to Deposit Account 02-2666 and please credit any excess fees to such account.

Respectfully submitted,

Dated: 12/30/04

Joni D. Stutman-Horn
Patent Attorney
Intel Corporation
Registration No. 42,173
(703) 633-6845

c/o Blakely, Sokoloff, Taylor &
Zafman, LLP
12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025-1026